**Monster Rolodex**

1. **Component state**

->state is a different object in memory , let me re-render it. And update the state/component.

->**shallow merge**(will search for the key and update the object) this object into your current state and give you new state of object

this.state= {monster1:{name:'Zuky'},

monster2:{name:'Linda'},

monster3:{name:'Frank'},

}

**//Instead of redefining the monsters again and again we will make an array that is**

this.state = {

monsters: [

{name:'Zuky'},

{name:'Linda'},

{name:'Frank'}, ]

};

**AND NOW WE USE MAP METHOD TO ITERATE**

<h1> {this.state.monster1.name} </h1>

<h1> {this.state.monster2.name} </h1>

<h1> {this.state.monster3.name} </h1>

**THAT IS**

{this.state.monsters.map((monster) =>{

return (

<div key={monster.id}>

<h1 >{monster.name}</h1>

</div>

);

1. **Component lifecycle:**

componentDidMount();

1. **Input search box component**
2. **Storing original data**
3. render(){
4. const filteredMonsters = this.state.monsters.filter((monster)=>{
5. return monster.name.toLocaleLowerCase().includes(this.state.searchString);
6. });
7. return (
9. <div className="App">
10. <input className='search-box' type='search' placeholder='search monsters' onChange={(event) =>{
11. console.log(event.target.value)
12. const searchString = event.target.value.toLocaleLowerCase();
14. this.setState(()=>{
15. return {searchString};
16. })
17. }}/>
18. {filteredMonsters.map((monster) =>{
19. return (
20. <div key={monster.id}>
21. <h1 >{monster.name}</h1>
22. </div>
23. );
24. })}
25. </div>
26. );
27. }
28. }

**5. Optimization**

->removing the extra re-rendering

1. onSearchChange = (event) =>{
2. console.log(event.target.value)
3. const searchString = event.target.value.toLocaleLowerCase();
4. this.setState(()=>{
   1. return {searchString};
5. });
6. };
7. render(){
8. const {monsters, searchString} = this.state;
9. const {onSearchChange} = this;
10. const filteredMonsters = monsters.filter((monster)=>{
    1. return monster.name.toLocaleLowerCase().includes(searchString);
11. });
12. return (
13. <div className="App">
14. <input className='search-box' type='search' placeholder='search monsters'
15. onChange= {onSearchChange} />
16. {filteredMonsters.map((monster) =>{
17. return (
18. <div key={monster.id}>
19. <h1 >{monster.name}</h1>
20. </div>
21. );
22. })}